# Clothing Entrepreneurship

**Sahil Upadhyay** 13503865 B13
**Shashank Gupta** 13503888 B13

# Problem Statement

Clothing Entrepreneurship

holds information regarding the raw material that goes to the factories, clothes sent to showrooms from factories, customer and employee records as well as the feedback records from the customers. It is used to store the information about the entire business setup.

We have designed a swing consisting of different jFrames for viewing the information for all the entities, inserting values into them, deleting a desired value if required and also for executing various queries running on our database. There is a login page which requires username and password to login. The flow of control from one JFrame to another is controlled using jButtons in java.

In our clothing Entrepreneurship project apart from entering information for various employees, customers, factories, showrooms, clothes and raw material through insert statements we can also view the information for these fields. We also have provision for the deletion of an employee record as well as a cloth record; in case it is sold to a customer it needs to be deleted from the showroom.

Our interface also includes certain queries to find out some vital information. For example, we have a query to find which factory transports clothes to maximum number of showrooms, what are the details of such showrooms, how to find clothes in a particular price range and also to find the maximum and average price of a cloth in a showroom. All these tasks have been achieved using stored procedures with and without cursors.

Lastly some triggers have been included to perform background tasks like providing hike in salary to an employee giving a good service and deleting cloth details if it is sold. In this way the project establishes a complete ownership and hence clothing entrepreneurship.

# Original EER to Relational Mapping

T1 (Fac_id, Owner, Name, Area, Workers, Address, Capacity, Cloth_id, Emp_id)

T2 (Raw_id, Quality, Price, Quantity)

T3 (Fac_id, Raw_id)

T4 (Raw_id, Show_id, Trans_id, Charges, Type)

T5 (City_id, City, Show_id, Fac_id)

T6 (Name, Email, Comments, Cust_id)

T7 (Cust_id, Paymode, Total, City, Email, Name, Cloth_id)

T8 (Cust_id, Phno)

T9 (Show_id, Name, Owner, Workers, Address, Quantity, Cloth_id, Emp_id)

T10 (Cloth_id, Size, Price, Material, Colour, Gender)

T11 (Emp_id, Name, Address, Salary, Age, Dept_id, Dept_name, Head)

# Functional Dependencies used for Normalization

Fac_id → Owner, Name, Workers, Address, Capacity

Fac_id → Raw_id, Cloth_id, Emp_id

Raw_id → Quality, Price, Quantity, Trans_id, Charges, Type

Raw_id → Show_id

City_id → City, Show_id, Fac_id

Cust_id → Name, Email, Comments, Paymode, Total

Cust_id → Cloth_id, Emp_id, City_id

Show_id → Name, Address, Workers, Quantity

Show_id → Cloth_id, Emp_id

Cloth_id → Price, Material, Size, Gender, Colour

Cloth_id → City_id

Emp_id → Name, Address, Salary, Age, Dept_id

Dept_id → Name, Head

Showroom_name → Owner

# Normalized relations till BCNF

T1 (Fac_id, Owner, Name, Workers, Address, Capacity)

T2 (Fac_id, Emp_id)

T3 (Fac_id, Cloth_id)

T4 (Raw_id, Quality, Price, Quantity)

T5 (Fac_id, Raw_id)

T6 (Raw_id, Show_id)

T7 (Raw_id, Trans_id, Charges, Type)

T8 (City_id, City)

T9 (City_id, Show_id)

T10 (City_id, Fac_id)

T11 (Name, Email, Comments, Cust_id)

T12 (Cust_id, Paymode, Email, Name, Total)

T13 (Cust_id, Cloth_id)

T14 (Show_id, Name, Address, Workers, Quantity)

T15 (Show_id, Cloth_id)

T16 (Show_id, Emp_id)

T17 (Cloth_id, Price, Material)

T18 (Cloth_id, Size, Gender)

T19 (Cloth_id, Colour)

T20 (Emp_id, Name, Address, Salary, Age, Dept_id)

T21 (Dept_id, Name, Head)

T22 (Cloth_id, City_id)

T23 (Cust_id, City_id)

T24 (Show_name, Owner)

# DDL Structure

```
SQL> create table t1
  2  (
  3  fac_id varchar2(10) primary key,
  4  owner varchar2(20),
  5  name varchar2(20),
  6  address varchar2(20),
  7  workers integer,
  8  capacity integer
  9  );

Table created.

SQL> create table t4
  2  (
  3  raw_id varchar2(10) primary key,
  4  quality varchar2(20),
  5  price integer,
  6  quantity integer
  7  );

Table created.

SQL> create table t12
  2  (
  3  cust_id varchar2(10) primary key,
  4  paymode varchar2(20),
  5  email varchar2(20),
  6  name varchar2(20),
```

```
 7  total integer
 8  );
```

Table created.

```
SQL> create table t14
 2  (
 3  show_id varchar2(10) primary key,
 4  name varchar2(20),
 5  address varchar2(20),
 6  workers integer,
 7  quantity integer,
 8  constraint fk_t14 foreign key (name) references t24(name)
 9  );
```

Table created.

```
SQL> create table t17
 2  (
 3  cloth_id varchar2(10) primary key,
 4  price integer,
 5  material varchar2(20)
 6  );
```

Table created.

```
SQL> create table t21
 2  (
 3  dept_id varchar2(10) primary key,
 4  name varchar2(20),
```

```
  5  head varchar2(20)
  6  );
```

Table created.

```
SQL> create table t20
  2  (
  3  emp_id varchar2(20) primary key,
  4  name varchar2(20),
  5  address varchar2(20),
  6  salary integer,
  7  age integer,
  8  dept_id varchar2(10),
  9  constraint fk_t20 foreign key (dept_id) references t21(dept_id)
 10  );
```

Table created.

```
SQL> create table t18
  2  (
  3  cloth_id varchar2(10),
  4  cloth_size varchar2(20),
  5  gender varchar2(20),
  6  constraint pk_t18 primary key (cloth_id,cloth_size,gender),
  7  constraint fk_t18 foreign key (cloth_id) references t17(cloth_id)
  8  );
```

Table created.

```
SQL> create table t19
```

```
  2  (
  3  cloth_id varchar2(10),
  4  colour varchar2(20),
  5  constraint pk_t19 primary key (cloth_id,colour),
  6  constraint fk_t19 foreign key (cloth_id) references t17(cloth_id)
  7  );
```

Table created.

```
SQL> create table t24
  2  (
  3  name varchar2(20) primary key,
  4  owner varchar2(20)
  5  );
```

Table created.

```
SQL> create table t8
  2  (
  3  city_id varchar2(10) primary key,
  4  city varchar2(20)
  5  );
```

Table created.

```
SQL> create table t16
  2  (
  3  show_id varchar2(10),
  4  emp_id varchar2(10),
  5  constraint pk_t16 primary key (show_id,emp_id),
```

```
  6  constraint fk_t16 foreign key (show_id) references t14(show_id),
  7  constraint fk_t16_1 foreign key (emp_id) references t20(emp_id)
  8  );
```

Table created.

```
SQL> create table t22
  2  (
  3  cloth_id varchar2(10),
  4  city_id varchar2(10),
  5  constraint pk_t22 primary key (cloth_id,city_id),
  6  constraint fk_t22 foreign key (cloth_id) references t17(cloth_id),
  7  constraint fk_t22_1 foreign key (city_id) references t8(city_id)
  8  );
```

Table created.

```
SQL> create table t23
  2  (
  3  cust_id varchar2(10),
  4  city_id varchar2(10),
  5  constraint pk_t23 primary key (cust_id,city_id),
  6  constraint fk_t23 foreign key (cust_id) references t12(cust_id),
  7  constraint fk_t23_1 foreign key (city_id) references t8(city_id)
  8  );
```

Table created.

```
SQL> create table t2
  2  (
```

```
  3  fac_id varchar2(10),
  4  emp_id varchar2(10),
  5  constraint pk_t2 primary key (fac_id,emp_id),
  6  constraint fk_t2 foreign key (fac_id) references t1(fac_id),
  7  constraint fk_t2_1 foreign key (emp_id) references t20(emp_id)
  8  );

Table created.

SQL> create table t3
  2  (
  3  fac_id varchar2(10),
  4  cloth_id varchar2(10),
  5  constraint pk_t3 primary key (fac_id,cloth_id),
  6  constraint fk_t3 foreign key (fac_id) references t1(fac_id),
  7  constraint fk_t3_1 foreign key (cloth_id) references t17(cloth_id)
  8  );

Table created.

SQL> create table t5
  2  (
  3  fac_id varchar2(10),
  4  raw_id varchar2(10),
  5  constraint pk_t5 primary key (fac_id,raw_id),
  6  constraint fk_t5 foreign key (fac_id) references t1(fac_id),
  7  constraint fk_t5_1 foreign key (raw_id) references t4(raw_id)
  8  );

Table created.
```

```
SQL> create table t6
  2  (
  3  raw_id varchar2(10),
  4  show_id varchar2(10),
  5  constraint pk_t6 primary key (raw_id,show_id),
  6  constraint fk_t6 foreign key (raw_id) references t4(raw_id),
  7  constraint fk_t6_1 foreign key (show_id) references t14(show_id)
  8  );

Table created.

SQL> create table t7
  2  (
  3  raw_id varchar2(10),
  4  trans_id varchar2(10),
  5  charges integer,
  6  type varchar2(20),
  7  constraint pk_t7 primary key (raw_id,trans_id),
  8  constraint fk_t7 foreign key (raw_id) references t4(raw_id)
  9  );

Table created.

SQL> create table t9
  2  (
  3  city_id varchar2(10),
  4  show_id varchar2(10),
  5  constraint pk_t9 primary key (city_id,show_id),
  6  constraint fk_t9 foreign key (city_id) references t8(city_id),
```

```
  7  constraint fk_t9_1 foreign key (show_id) references t14(show_id)
  8  );
```

Table created.

```
SQL> create table t10
  2  (
  3  city_id varchar2(10),
  4  fac_id varchar2(10),
  5  constraint pk_t10 primary key (city_id,fac_id),
  6  constraint fk_t10 foreign key (city_id) references t8(city_id),
  7  constraint fk_t10_1 foreign key (fac_id) references t1(fac_id)
  8  );
```

Table created.

```
SQL> create table t11
  2  (
  3  cust_id varchar2(10),
  4  name varchar2(20),
  5  email varchar2(20),
  6  comments varchar2(20),
  7  constraint fk_t11 foreign key (cust_id) references t12(cust_id),
  8  constraint pk_t11 primary key (name,email)
  9  );
```

Table created.

```
SQL> create table t13
  2  (
```

```
  3  cust_id varchar2(10),
  4  cloth_id varchar2(10),
  5  constraint pk_t13 primary key (cust_id,cloth_id),
  6  constraint fk_t13 foreign key (cust_id) references t12(cust_id),
  7  constraint fk_t13_1 foreign key (cloth_id) references t17(cloth_id)
  8  );
```

Table created.

```
SQL> create table t15
  2  (
  3  show_id varchar2(10),
  4  cloth_id varchar2(10),
  5  constraint pk_t15 primary key (show_id,cloth_id),
  6  constraint fk_t15 foreign key (show_id) references t14(show_id),
  7  constraint fk_t15_1 foreign key (cloth_id) references t17(cloth_id)
  8  );
```

Table created.

# DML Commands

```
mysql> insert into t1 values('1001','Avish Kumar','Cloth Emporium',50,'Delhi',1000);
Query OK, 1 row affected (0.06 sec)
mysql> insert into t4 values('101','Silk',250,1000);
Query OK, 1 row affected (0.06 sec)
mysql> insert into t5 values('1001','101');
Query OK, 1 row affected (0.06 sec)
mysql> insert into t14 values('S1','South Ex',10,'Fashion Club',250);
Query OK, 1 row affected (0.06 sec)
mysql> insert into t7 values('C1',500,'Woolen');
Query OK, 1 row affected (0.06 sec)
mysql> insert into t19 values('C1','Yellow');
Query OK, 1 row affected (0.06 sec)
mysql> insert into t15 values('S1','C1');
Query OK, 1 row affected (0.06 sec)
mysql> insert into t24 values('Fashion Club','Annapoorna');
Query OK, 1 row affected (0.06 sec)
mysql> insert into t20 values('1','Ajay','Lucknow',12000,22,'D2');
Query OK, 1 row affected (0.06 sec)
mysql> insert into t21 values('D2','Salesman','Prakhar');
Query OK, 1 row affected (0.06 sec)
mysql> insert into t16 values('S1','1');
Query OK, 1 row affected (0.06 sec)
mysql> insert into t8 values('City_1','Delhi');
Query OK, 1 row affected (0.06 sec)
mysql> insert into t22 values('C1','City_1');
Query OK, 1 row affected (0.06 sec)
mysql> insert into t2 values('1001','1');
```

Query OK, 1 row affected (0.06 sec)

mysql> insert into t3 values('1001','C1');

Query OK, 1 row affected (0.06 sec)

mysql> insert into t9 values('City_1','S1');

Query OK, 1 row affected (0.06 sec)

mysql> insert into t10 values('City_1','1001');

Query OK, 1 row affected (0.06 sec)

# Code for Triggers

1. Trigger to insert values into table t19 whenever a new cloth_id is entered into table t17.

```
mysql> delimiter $$
mysql> create trigger trig
   -> after insert on t17
   -> for each row
   -> begin
   -> insert into t18 values(new.cloth_id,'S','MALE');
   -> insert into t18 values(new.cloth_id,'M','MALE');
   -> insert into t18 values(new.cloth_id,'L','MALE');
   -> insert into t18 values(new.cloth_id,'S','FEMALE');
   -> insert into t18 values(new.cloth_id,'M','FEMALE');
   -> insert into t18 values(new.cloth_id,'L','FEMALE');
   -> end $$
Query OK, 0 rows affected (0.13 sec)
```

2. Trigger to pay a hike of 500 rupees to an employee whose emp_id is mentioned in the feedback.

```
mysql> delimiter $$
mysql> create trigger trig1
   -> after insert on t11
   -> for each row
   -> begin
   -> update t25
   -> set salary=salary+500
   -> where emp_id=new.emp_id;
   -> end$$
Query OK, 0 rows affected (0.58 sec)
```

3. Trigger to delete cloth information from tables t18 and t19 whenever a cloth_id is deleted from table t17.

```
mysql> delimiter $$
mysql> create trigger trig2
   -> before delete on t17
   -> for each row
   -> begin
   -> delete from t18 where cloth_id=old.cloth_id;
   -> delete from t19 where cloth_id=old.cloth_id;
   -> end$$
Query OK, 0 rows affected (0.52 sec)
```

4. Trigger to trigger salary of an employee making it 15000 on insertion of a new record.

```
mysql> delimiter $$
mysql> create trigger trig3
   -> after insert on t20
   -> for each row
   -> begin
   -> insert into t25
   -> values(new.emp_id,15000);
   -> end $$
Query OK, 0 rows affected (0.09 sec)
```

# Code for Procedures

## 1. Procedure to display maximum and average price of cloth in any showroom.

```
mysql> delimiter $$
mysql> create procedure pro (in s_id varchar(20))
   -> begin
   -> select max(prize),avg(prize)
   -> from t15 join t17
   -> on(t15.cloth_id=t17.cloth_id)
   -> group by show_id
   -> having show_id=s_id;
   -> end$$
Query OK, 0 rows affected (0.08 sec)
```

## 2. Procedure to find clothes within a range of price.

```
mysql> delimiter $$
mysql> create procedure pro1 (in pmax integer,in pmin integer)
   -> begin
   -> select t17.cloth_id,material,colour
   -> from t17 join t19
   -> on(t17.cloth_id=t19.cloth_id)
   -> where prize<pmax and prize>pmin;
   -> end$$
Query OK, 0 rows affected (0.00 sec)
```

## 3. Procedure to find which factory transports clothes to maximum number of showrooms.

```
mysql> delimiter $$
mysql> create procedure pro2()
   -> begin
   -> create view view1 as
```

```
-> select count(distinct show_id) show_count
-> from t3 join t15
-> on(t3.cloth_id=t15.cloth_id)
-> group by fac_id;
-> create view view2 as
-> select max(show_count) max_show_count from view1;
-> select t3.fac_id,fac_name,count(distinct show_id)
-> from t1 join t3
-> on(t1.fac_id=t3.fac_id)
-> join t15
-> on(t3.cloth_id=t15.cloth_id)
-> group by fac_id
-> having count(distinct show_id) in(select max_show_count from view2);
-> end$$
```

Query OK, 0 rows affected (0.00 sec)

# Code for Procedures using Cursor

## 1. Cursor to find all the showrooms in a particular city.

```
mysql> delimiter $$
mysql> create procedure curs(in cityid varchar(20))
   -> begin
   -> declare sid varchar(20);
   -> declare exit_loop boolean;
   -> declare curt cursor for
   -> select show_id from t9
   -> where city_id=cityid;
   -> declare continue handler for not found set exit_loop=true;
   -> open curt;
   -> tloop: loop
   -> fetch curt into sid;
   -> if exit_loop then
   -> close curt;
   -> leave tloop;
   -> end if;
   -> select show_id,name from t14 where show_id=sid;
   -> end loop;
   -> end $$
Query OK, 0 rows affected (0.00 sec)
```

## 2. Cursor to find which factory transfers clothes to which all showrooms.

```
mysql>delimiter $$
mysql> create view v12 as
   -> select fac_id,show_id
   -> from t3 join t15
   -> on(t3.cloth_id=t15.cloth_id);
```

```
   -> $$
Query OK, 0 rows affected (0.09 sec)


mysql> create procedure curs1(in facid varchar(20))
   -> begin
   -> declare sid varchar(20);
   -> declare exit_loop boolean;
   -> declare curt1 cursor for
   -> select distinct show_id from v12 where fac_id=facid;
   -> declare continue handler for not found set exit_loop=true;
   -> open curt1;
   -> tloop: loop
   -> fetch curt1 into sid;
   -> if exit_loop then
   -> close curt1;
   -> leave tloop;
   -> end if;
   -> select show_id,name from t14 where show_id=sid;
   -> end loop;
   -> end $$
Query OK, 0 rows affected (0.00 sec)
```