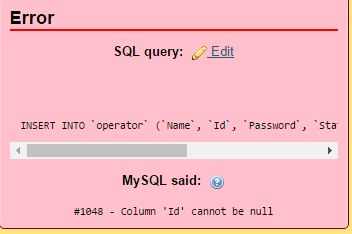
# TRIGGERS

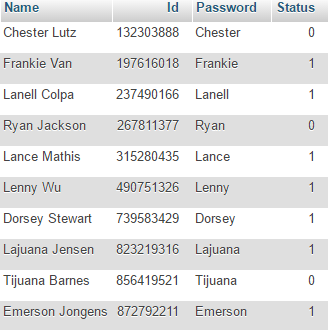
// Don't use these triggers on the database created for phase 2.......the foreign key constraint will take precedent and second trigger will not be activated

***CREATE TRIGGER `check1` BEFORE INSERT ON `operator` FOR EACH ROW begin if new.Status >1 OR new.Status < 0 then set new.Id = NULL; end if; end;***

The status field of the operator table should either be 0 (Currently not available i.e. in line with some customer) or 1 (currently available i.e. ready to speak to a new customer). A constraint is set during table creation which ensures that the value being stored is single digit. But we can still any value from 2 to 9 other than 0 or 1. This trigger will make sure the value is only 0 or 1. If the value is other than them, the Id field is set to NULL which results in query failing primary key constraint.



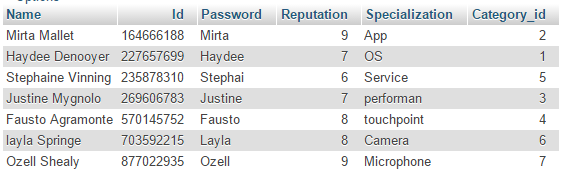
Error message when we try to execute a query with status value other than 0 or 1.

The picture shows the data in the table after execution of query.

***CREATE TRIGGER `checkid` BEFORE INSERT ON `resolved` FOR EACH ROW BEGIN DECLARE sid int(1);SELECT COUNT(1) INTO sid FROM specialist WHERE Id = NEW.Specialist\_id; if sid = 0 then SET new.Specialist\_id = NULL; end if; END***

The above trigger ensures the foreign key constraint. The specialist\_id in resolved table should be in the specialist table. Whenever we don’t find the specialist\_id in the specialist table we make the specialist\_id in the resolved table NULL which fails the not null constraint.

Data in the specialist table



Error message when executing the query.

