Normalisation

UniversalRelation =
{name, user_ID, password, email, c_ID, c_homeaddr, c_phoneNo, p_ID, p_scale, p_officeaddr, p_phoneNo, p_multiplier, p_PAN, p_GST, adm_ID, adm_join_date, req_ID, req_weight, req_size, req_speed, req_dist, req_type, order_ID, weight, size, type, speed, start_date, status, dist, quote_ID, quote_amt, quote_speed, emp_ID, emp_phoneNo, emp_email, emp_salary}

Functional Dependencies

user_ID ➔ name, password, email
c_ID ➔ c_homeaddr, c_phoneNo
p_ID ➔ p_scale, p_officeaddr, p_phoneNo, p_multiplier, p_PAN, p_GST
adm_ID ➔ adm_join_date
req_ID ➔ req_weight, req_size, req_dist, req_speed, req_type
quote_ID ➔ quote_amt, quote_speed
order_ID➔ weight, speed, size, dist, type
emp_ID ➔ emp_email, emp_salary
order_ID,c_ID ➔start_date
order_ID,p_ID➔status

Normalising to First Normal Form (1NF):

A relational schema R is in first normal form if the domains of all attributes if R are atomic, i.e it disallows composite attributes, multivalued attributes, and nested relations.

Thus removing emp_phoneNo from UniversalRelation and creating following relations.

R1 (name, user_ID, password, email, c_ID, c_homeaddr, c_phoneNo, p_ID, p_scale, p_officeaddr, p_phoneNo, p_multiplier, p_PAN, p_GST, adm_ID, adm_join_date, req_ID, req_weight, req_size, req_speed, req_dist, req_type, order_ID, weight, size, type, speed, start_date, status, dist, quote_ID, quote_amt, quote_speed, emp_ID, emp_email, emp_salary)

R2 (emp_ID, emp_phoneNo)

Normalising to Second Normal Form (2NF):

2NF is based on the concept of Full Functional Dependency.
A relation schema R is in 2NF if it is in 1NF form and every non-prime attribute A in R is fully functionally dependent on the primary key of R.

In R1, we observe the existence of some partial functional dependencies. We see that the attributes are fully functionally dependent on the primary keys user_ID, c_ID, p_ID, adm_ID, req_ID, quote_ID, order_ID, and emp_ID respectively.So as to preserve the integrity of the second normal form we decompose R1 into the following relations.

R1_A (user_ID, name, password, email);
R1_B (c_ID, c_homeaddr, c_phoneNo);
R1_C (p_ID, p_scale, p_officeaddr, p_phoneNo, p_multiplier, p_PAN, p_GST);
R1_D (adm_ID, adm_join_date);
R1_E (req_ID, c_ID, req_weight, req_size, req_speed, req_dist, req_type);
R1_F (quote_ID, p_ID,quote_amt, quote_speed);
R1_G (order_ID,c_ID,p_ID, weight, size, type, speed, start_date, status, dist)
R1_H (emp_ID, emp_email, emp_salary);

R2 (emp_ID, emp_phoneNo);

Normalising to Third Normal Form (3NF):

A relation is in the third normal form, if there is no transitive dependency for non-prime attributes as well as it is in the second normal form. A relation is in 3NF if at least one of the following conditions holds in every non-trivial functional dependency X –> Y.

- *X is a super key.*
- *Y is a prime attribute (each element of Y is part of some candidate key).*

Since there are no transitive dependencies in any of the relations, all relations are already in Third Normal Form (3NF).

Normalised Relational Schema in 3NF:

R1 (user_ID, name, password, email);
R2 (emp_ID, emp_phoneNo);
R3 (c_ID, c_homeaddr, c_phoneNo);
R4 (p_ID, p_scale, p_officeaddr, p_phoneNo, p_multiplier, p_PAN, p_GST);
R5 (adm_ID, adm_join_date);
R6 (req_ID, c_ID, req_weight, req_size, req_speed, req_dist, req_type);
R7 (quote_ID, p_ID,quote_amt, quote_speed);
R8 (order_ID,c_ID,p_ID, weight, size, type, speed, start_date, status, dist)
R9 (emp_ID, emp_email, emp_salary);