

Normal

Supplier

<u>sup.id</u>	sname	item	qty	amt	date ordered	date supplied

Problems starts at the time of implementation & int design.

~~Insert anomaly~~ Can't insert multiple items supplied by same supplier.
~~Delete anomaly~~ Can't delete a supplier ^{order} who supplied faulty items, as his whole history will get deleted.
~~Update anomaly~~ Can't update a particular field as that will alter old info too.

If we have 6 tables in a DB, and we normalize ^{even} two tables to 3NF, our database is normalized.

1NF → ~~needs atomicity~~ allow redundant values
 2NF → ~~no~~ happens due to composite primary keys
 3NF → ~~no~~ no transitive

2NF

student (20000)

s_num c_num sname cname int. room

composite primary key

breakdown the table to remove composite key

3NF

Transitive dependency

3NF

lossless

4NF

accept multiple data with same fields.
this is not possible in Oracle & SQL server.
as 'array' is provided with DBMS system

5NF

If after decomposing & joining back the tables,
we get the original table.

##

PL/SQL → for Oracle

Programmable logic - SQL

stored procedure in MySQL & SQL server

PL/SQL sends entire block of statements to the Oracle engine
at one time.

Provides facilities of conditional nesting, branching and
looping.

Error handling.

Q- write a PL/SQL procedure to calculate the area of a circle.

The radius should be taken as input. Increase radius by 4
constant value to find area of circle with different radii.

~~declare~~ area (radius, num, pi num, area num);

declare

pi constant number (4, 2) := 3.14;

radius number (5);

area number (14, 2);

begin

radius := 3;

loop

area := pi * power(radius, 2);

dbms_output.put_line(area);

radius := radius + 1;

exit when radius > 5;

end loop;

end;

- PL/SQL is a block-structured language.
- The basic units (procedures, functions and anonymous blocks) that make up a PL/SQL program are logical blocks, which can contain any no. of nested sub-blocks.
- Each logical block corresponds to a problem or sub-problem to be solved.