Normalisation - Sample Solutions

Question A

Assume a patient [illegible] once per day
If using Oracle ha[illegible]ribute containing both date and time

1.

INSERT anomaly:
Can't insert a dentist until they have a patient appointment

DELETE anomaly:
When the last existing record of an appointment for a dentist is deleted, the dentist's details are lost

UPDATE anomaly:
If a dentist's details are to be updated eg: change of name, multiple rows need to be updated

2.
(1)

UNF:
APPOINTMENT(staffno, dentistname, {patno, patname, appointment, surgeryno})

(2)

UNF:
DENTIST(staffno, dentistname, {patno, patname, appointment, surgeryno})

(3)

UNF:
DENTIST(staffno, dentistname, {patno, patname, {appointment, surgeryno}})

3.

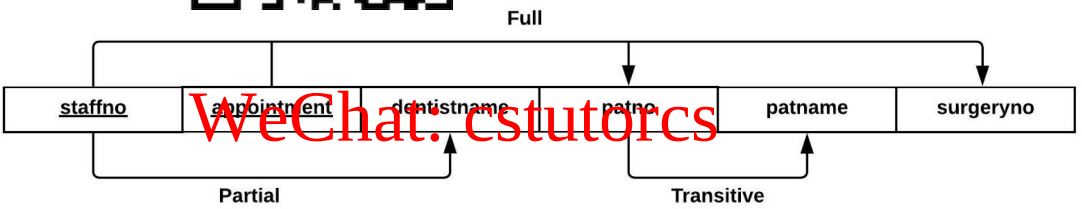
UNF:

APPOINTMENT(staffno, appointment, dentistname, patno, patname, , surgeryno)

Using Simple De[illegible]:

1NF:

APPOINTMENT(staffno, appointment, dentistname, patno, patname, , surgeryno)



OR

staffno, appointment -> patno, surgeyno (FULL)
staff_no -> dentistname PARTIAL
patno -> patname TRANSITIVE

2NF:

DENTIST (staffno, dentistname)
APPOINTMENT (staffno, appointment, patno, patname, surgeryno)

3NF:

DENTIST (staffno, dentistname)
APPOINTMENT (staffno, appointment, patno, surgeryno)
PATIENT (patno, patname)

4.

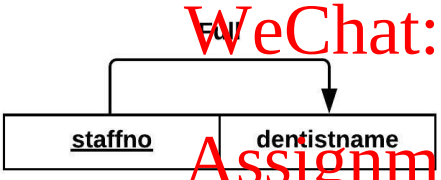
UNF:

DENTIST (staffn, patname, appointment, surgeryno))

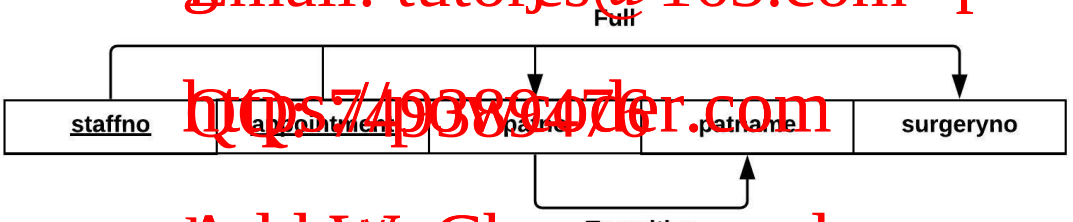
Using Simple De K:

1NF:

DENTIST (staffn
APPOINTMENT(*staffno*, appointment, patno,patname,surgeryno)



OR
staffno -> dentistname FULL



OR
staffno, appointment -> patno, surgeryno FULL
patno -> patname TRANSITIVE

2NF:

There is no partial dependency.

DENTIST (staffno, dentistname)
APPOINTMENT(*staffno*, appointment, patno,patname,surgeryno)

3NF:

DENTIST (staffno, dentistname)
APPOINTMENT(*staffno*, appointment, *patno*,surgeryno)
PATIENT(patno,patname)

Question B

STEP 1: NORMALISATION:

Take each form o and list it as a UNF relation, then normalise through 1NF, 2NF and 3N malisation data until you have completed all the normalisations.

APPROVED UN

UNF
UNIT (unit_no, unit_name, unit_desc, unit_value)

1NF
UNIT (unit_no, unit_name, unit_desc, unit_value)

OR
unit_no -> unit_name, unit_desc, unit_value FULL

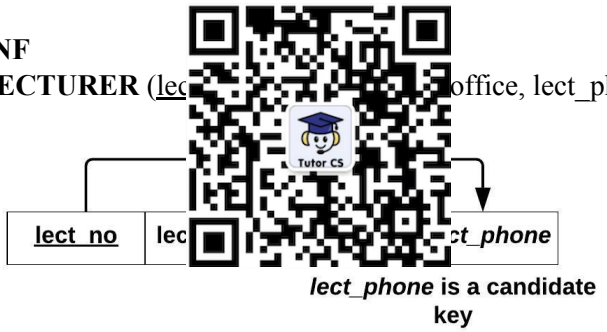
2NF
UNIT (unit_no, unit_name, unit_desc, unit_value)

3NF
UNIT (unit_no, unit_name, unit_desc, unit_value)

LECTURER REPORT

UNF
LECTURER (lect_no, lect_name, lect_office, lect_phone (unit_no, unit_name))

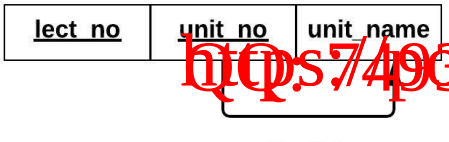
1NF
LECTURER (lec office, lect_phone)



There is no transitive dependency here related to lect_phone as lect_phone is a candidate key - transitive dependency is about the *removal of non-key dependencies* ie. dependencies between non-key attributes (lect_phone is not a non-key attribute)

OR
lect_no -> lect_name, lect_office, lect_phone FULL

ADVISES (lect_no, unit_no, unit_name)

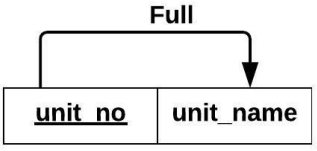


OR
unit_no -> unit_name PARTIAL

2NF
LECTURER (lect_no, lect_name, lect_office, lect_phone)

ADVISES (lect_no, unit_no)

UNIT (unit_no, unit_name)



OR
unit_no -> unit_name FULL

3NF

LECTURER (lect_no, lect_name, lect_office, lect_phone)

(lect_phone is a candidate key and hence transitive dependencies are not present)

ADVISES (lect_r

UNIT (unit_no, u

STUDENT REP(

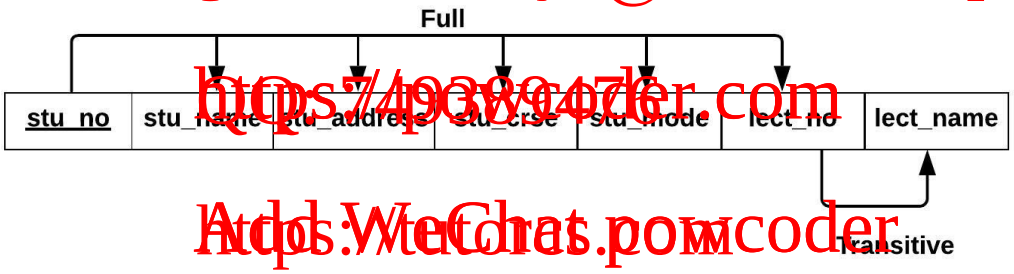
UNF

STUDENT (stu_no, stu_name, stu_address, stu_crse, stu_mode, lect_no, lect_name (unit_no, unit_name, yr_sem, grade))

Note: replacement of mentor details with lecturer details - a mentor is a lecturer - this prevents the introduction of synonyms (attributes with different names but representing the same thing)

1NF

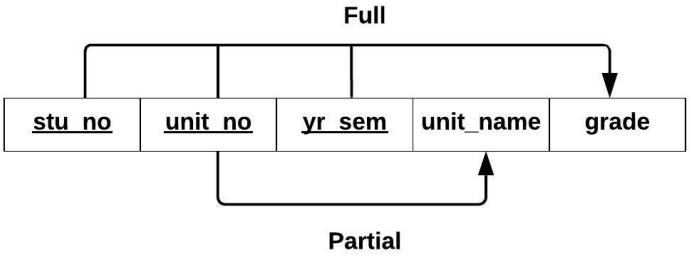
STUDENT (stu_no, stu_name, stu_address, stu_crse, stu_mode, lect_no, lect_name)



OR

stu_no -> stu_name, stu_address, stu_crse, stu_mode, lect_no FULL
lect_no -> lect_name TRANSITIVE

AC-REC (stu_no, unit_no, yr_sem , unit_name, grade)



OR

stu_no, unit_no, yr_sem -> grade
unit_no -> unit_name PARTIAL

2NF

STUDENT (stu_no, stu_name, stu_address, stu_crse, stu_mode, lect_no, lect_name)

AC-REC (stu_no, unit_no, yr_sem, grade)

UNIT (unit_no, u

OR

unit_no -> unit_name FULL

3NF

STUDENT (stu_no, stu_name, stu_address, stu_crse, stu_mode, lect_no)

LECTURER (lect_no, lect_name)

OR

lect_no -> lect_name

AC-REC (stu_no, unit_no, yr_sem, grade)

UNIT (unit_no, unit_name)

COLLECTED 3 NF Relations:

1. UNIT (unit_no, unit_name, unit_desc, unit_value)

2. LECTURER (l_office, lect_phone)

3. ADVISES (lect

4. UNIT (unit_no,

5. STUDENT (stu dress, stu_crse, stu_mode, lect_no)

6. LECTURER (lect_no, lect_name)

7. AC-REC (stu_no, unit_no, yr_sem, grade)

8. UNIT (unit_no, unit_name)

STEP 2: ATTRIBUTE SYNTHESIS

Join together relations, which have an **identical** PK – ie. represent the same entity:

1. 4. & 8.
UNIT (unit_no, unit_name, unit_desc, unit_value)

2. & 6.
LECTURER (lect_no, lect_name, lect_office, lect_phone)

3.
ADVISES (lect_no, unit_no)

5.
STUDENT (stu_no, stu_name, stu_address, stu_crse, stu_mode, lect_no)

7.
AC-REC (stu_no, unit_no, yr_sem, grade)

Logical Model