Normalisation - Sample Solutions

Question A

Assume a patient can only see a dentist once per day
If using Oracle have an appointment attribute 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 appointment for a dentist is deleted, the dentists details are lost

UPDATE anomaly:
If a dentists 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}})

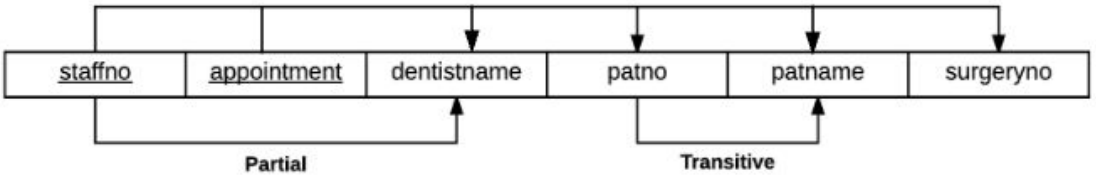
3.

UNF:

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

Using Simple Definition - based on PK:

1NF:
APPOINTMENT (staffno, appointment , dentistname, patno, patname, , surgeryno)



OR (as an alternative notation)
staffno, appointment -> dentistname, patno, patname, surgeyno
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)

Using General Definition - based on candidate Keys:
1NF:

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

staffno, appointment -> dentistname, patno, patname, surgeyno
staff_no -> dentistname PARTIAL based on PK
patno -> patname PARTIAL based on CK patno, appointment

2NF:

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

3NF:
DENTIST (staffno, dentistname)
PATIENT (patno, patname)
APPOINTMENT (*staffno*, appointment , patno, surgeryno)

4.

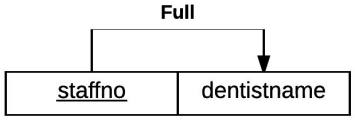
UNF:

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

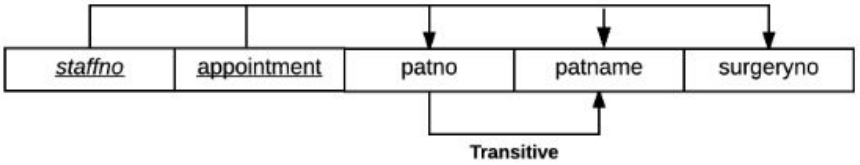
Using Simple Definition - based on PK:

1NF:

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



OR
staffno -> dentistname FULL



OR
staffno, appointment -> patno, patname, 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)

Using General Definition - based on candidate Keys:
1NF:

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

staffno -> dentistname
staffno, appointment -> patno, patname, surgeyno
patno -> patname PARTIAL based on CK patno, appointment

2NF:

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

3NF:

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

Question B

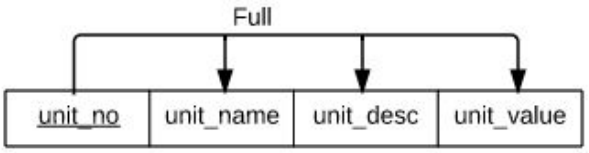
STEP 1: NORMALISATION:

Take each form on a *form-by-form* basis and list it as a UNF relation, then normalise through 1NF, 2NF and 3NF. Do not pool the normalisation data until you have completed all the normalisations.

APPROVED UNITS REPORT

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

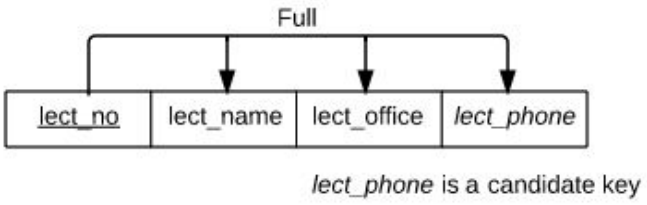
2NF
UNIT (unit_no, unit_name, unit_desc, unit_value)

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

LECTURER REPORT

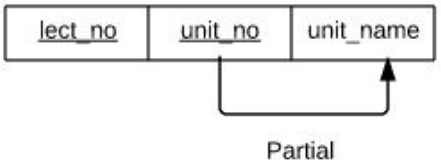
UNF
LECTURER (lect_no, lect_name, lect_office, lect_phone (unit_no, unit_name))

1NF
LECTURER (lect_no, lect_name, lect_office, lect_phone)



OR
lect_no -> lect_name, lect_office, lect_phone FULL

ADVISES (lect_no, unit_no, unit_name)

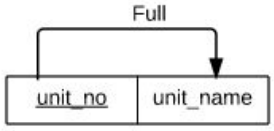


OR
lect_no, unit_no -> unit_name
unit_no -> unit_name PARTIAL

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_no, unit_no)

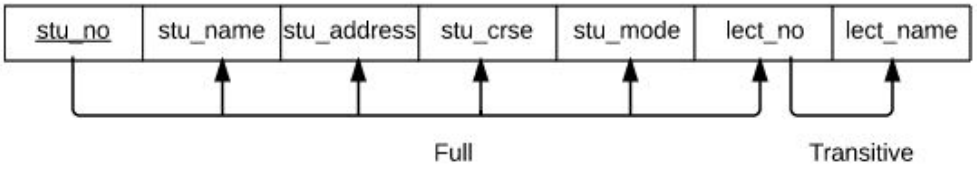
UNIT (unit_no, unit_name)

STUDENT REPORT

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

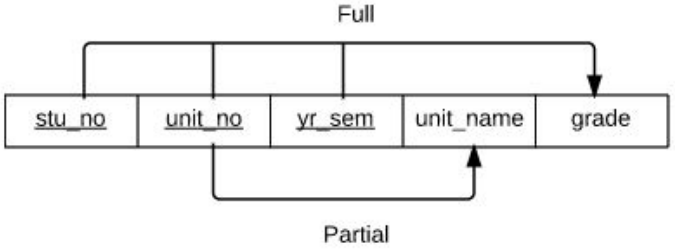
Note replacement of mentor details with lecturer details - a mentor is a lecturer - this prevents the introduction of synonms (attributes with different names but representing the same thing)

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, lect_name FULL
lect_no -> lect_name TRANSITIVE

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

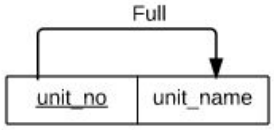


OR
stu_no, unit_no, yr_sem -> unit_name, grade
unit_no -> unit_name PARTIAL

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, unit_name)



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 (lect_no, lect_name, lect_office, lect_phone)

3. ADVISES (lect_no, unit_no)

4. UNIT (unit_no, unit_name)

5. STUDENT (stu_no, stu_name, stu_address, 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)