Normalization

1. Using the INVOICE table structure shown in the table, draw its dependency diagram and identify all dependencies (including all partial and transitive dependencies). You can assume that the table does not contain repeating groups and that any invoice number may reference more than one product. You can also assume that any given product is supplied by a single vendor, but a vendor can supply many products. Therefore, it is proper to conclude that the following dependency exists:

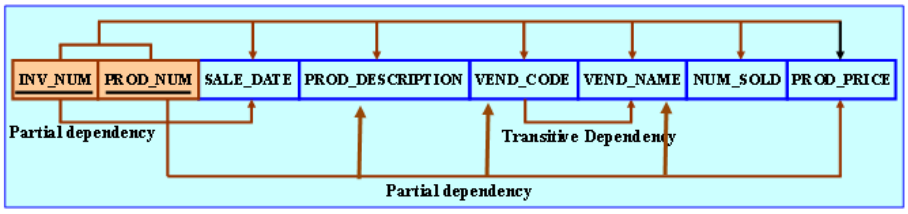
   PROD_NUM → PROD_DESCRIPTION, PROD_PRICE, VEND_CODE, VEND_NAME

   (*Hint*: This table uses a composite primary key.)

Table 1 Sample INVOICE Records

Attribute Name	Sample Value	Sample Value	Sample Value	Sample Value	Sample Value
INV_NUM	211347	211347	211347	211348	211349
PROD_NUM	AA-E3422QW	QD-300932X	RU-995748G	AA-E3422QW	GH-778345P
SALE_DATE	15-Jan-2004	15-Jan-2004	15-Jan-2004	15-Jan-2004	16-Jan-2004
PROD_LABEL	Rotary sander	0.25-in. drill bit	Band saw	Rotary sander	Power drill
VEND_CODE	211	211	309	211	157
VEND_NAME	NeverFail, Inc.	NeverFail, Inc.	BeGood, Inc.	NeverFail, Inc.	ToughGo, Inc.
QUANT_SOLD	1	8	1	2	1
PROD_PRICE	$49.95	$3.45	$39.99	$49.95	$87.75

Problem 1 Solution



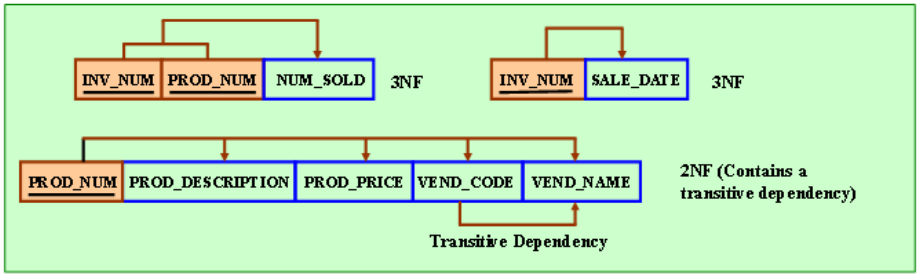
2. Using the initial dependency diagram drawn in Problem 1, remove all partial dependencies, draw the new dependency diagrams, and identify the normal forms for each table structure you created.

   (*Hint*: Your actions should produce three dependency diagrams.)

Commented [JN1]: Normalization: Functional Dependency Summary Steps (youtube.com)

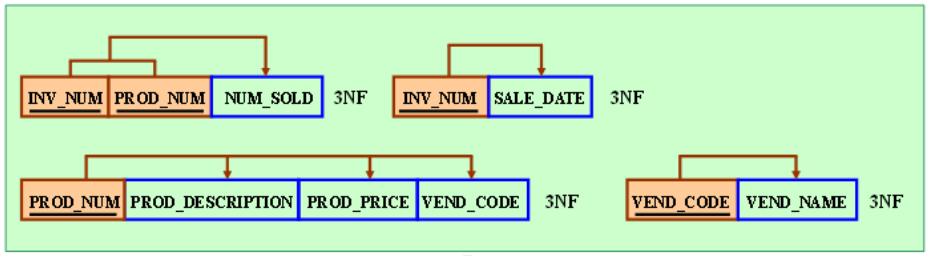
Commented [JN2]: I can know Vendor name, by just looking at Vendor Code

Problem 2 Solution



3. Using the table structures you created in Problem 2, remove all transitive dependencies, draw the new dependency diagrams, and identify the normal forms for each table structure you created.

Problem 3 Solution



4. Dream Sdn Bhd is a tour agency is using traditional file-based system to manage and maintain all records of their customers' bookings. TourPackages Table shown below is an example of data file stored in their system.

TourPackages Table

Package Code	Package Desc	Price (RM)	Tour Guide	T_Guide Contact	Cust No	Cust Name	Cust Phone	Depart Date
C123	8 days Shanghai	3800	Andy	011124563	M54	Mary W	016112312	05-07-15
					S11	Su Si	012456567	05-07-15
					G05	Gary Lim	012456678	05-07-15
					Y02	Yumiko	011225436	05-07-15
T007	10 days Turkey	6200	Sally	012569563	B12	Braham	011235645	15-08-15
E236	10 days UK, France	12000	Brian	012870900	M54	Mary W	016112312	08-04-15
					S11	Su Si	012456567	08-04-15
					S54	Susan	017235646	08-04-15
					N37	Natalie	012456465	08-04-15
					B77	Kate S	016458965	08-04-15
J560	6 days Tokyo, Osaka	4300	Yuki	016258836	Y02	Yumiko	011225436	02-05-15
					A39	Andrew	013455667	02-05-15

(a) Based on the sample data given in the table, discuss each of the following data anomalies with a specific example:

- Insertion anomaly
- Modification anomaly
- Deletion anomaly

(b) Normalize the table given below to a set of third normal form (3NF) relations using DBDL format.

Insertion anomaly – occurs when there is new customer to be inserted to the table. This new customer's record cannot be inserted without customer signing-up a package because PackageCode is a primary key which cannot contain null value..

Deletion anomaly – occurs when the package (PackageCode : T007) is cancelled and the record is removed from the table. The deletion of this package record may lead to the deletion of CustNo: B12's record from the table.

Modification anomaly – occurs when the departure date for PackageCode: E236 is changed, all the corresponding records must be updated accordingly, failing which may cause data inconsistency in the table.

1NF – Removal of repeating groups

Package (PackageCode, PackageDesc, Price, TourGuide, T_GuideContact)
BookingDetail (*PackageCode, CustNo, CustName, CustPhone, DepartDate)

2NF – Removal of partial dependencies

Package (PackageCode, PackageDesc, Price, TourGuide, T_GuideContact)
Customer (CustNo, CustName, CustPhone)
BookingDetail (*PackageCode,* CustNo, DepartDate)

3NF – Removal of transitive dependencies

Package (PackageCode, PackageDesc, Price, *TourGuide)
TourGuide (TourGuide, T_GuideContact)
Customer (CustNo, CustName, CustPhone)
BookingDetail (*PackageCode,* CustNo, DepartDate)

Normal Form

Commented [JN3]: •**First Normal Form (1NF):** This is the most basic level of normalization. In 1NF, each table cell should contain only a single value, and each column should have a unique name. The first normal form helps to eliminate duplicate data and simplify queries.
•**Second Normal Form (2NF):** 2NF eliminates redundant data by requiring that each non-key attribute be dependent on the primary key. This means that each column should be directly related to the primary key, and not to other columns.
•**Third Normal Form (3NF):** 3NF builds on 2NF by requiring that all non-key attributes are independent of each other. This means that each column should be directly related to the primary key, and not to any other columns in the same table.

5. Lovely Pet Clinic stores health history records of its customers' pets as shown in the table below:

VisitNo	petID	petName	owner	visitDate	treatID	treatDesc	dosage	fee
A101	520	Max	Niny	13-1-2016	V01	Rabies Vaccination	0.5 ml	140
A102	725	Ginger	Coco	21-2-2016	G20	LeptoVaccination	2 ml	150
					T12	Eye Treatment	100 mg	90
A103	211	Fatfat	Niny	13-3-2016	V01	Rabies Vaccination	1 ml	180
					T10	Treat wound	500mg	50
					T05	Heart worm medicine	110mg	120
A104	913	Circle	Coco	21-1-2016	V08	Tetanus Vaccination	1 ml	100

(a) Based on the sample data above, explain the meaning of functional dependency.

(b) Based on the sample data given in the table, discuss each of the following data anomalies with a specific example:

- Insertion anomaly
- Modification anomaly
- Deletion anomaly

(c) Normalize the table given below to a set of third normal form (3NF) relations using DBDL format.

Functional dependency is a relationship that exists when one (set of) attribute(s) uniquely determines the value of another(set of) attribute(s).

If student is a relation with attributes studentID and studentName, a**functional dependency**between the attributes is represented as studentID->studentName, which specifies studentName is**functionally**dependent on studentID.

Insertion anomaly
Inability to add new records without adding redundant data

Eg:A user needs to repeatedly enter petName, petName andowner when the petFatfat revisits the clinic. The user has to ensure that the data entered is uniform; otherwise the data will be inconsistent.

Modification anomaly
Inability to accurately and efficiently maintain data. A change to an attribute's values in one record requires changes in multiple records.

Eg: When the treatDesc for treatIDV01 is changed, user has to ensure that changes to the value in each row of thattreatIDis uniform; otherwise the data will be inconsistent.

Deletion anomaly
Inability to delete unwanted data without deleting data that you need to retain.
Eg: Suppose the clinic is no longer offering Tetanus Vaccination, the particular record should be removed from the table. As a result the data about the pet Circlewill be lost.

1NF: (3 marks)
Visit (visitNo, petID, petName, owner,visitDate)
PetTreatment(*visitNo, treatID, treatDesc, dosage, fee)
OR
Visit (visitNo, petID, petName, owner, visitDate, treatID, treatDesc, dosage, fee)

2NF: (4 marks)
Visit (visitNo, petID, petName, owner, visitDate)
PetTreatment (*visitNo, *treatID, dosage, fee)
Treatment(treatID, treatDesc)

3NF: (5 marks)
Pet (petID, petName, owner)
Visit (visitNo, *petID, visitDate)
PetTreatment (*visitNo, *treatID, dosage, fee)
Treatment(treatID, treatDesc)

6.The table shown below contains facts about patients, health care providers, patients' visits to a clinic, and diagnoses made by health care providers.

VisitNo	VisitDate	PatNo	PatAge	PatCity	PatZip	ProvNo	ProvSpecialty	Diagnosis
V20030	13/1/2005	P1	35	Lahat	31600	D1	Internist	Ear Infection
V20030	13/1/2005	P1	35	Lahat	31600	D2	Nurse Practioner	Influenza
V82020	20/1/2005	P3	20	Tronoh	32200	D2	Nurse Practioner	Pregnancy
V73220	18/1/2005	P2	62	Taiping	34500	D3	Cardiologist	Murmur

Normalize the table into a set of 3NF relations using DBDL format. Hint: The zip code of the patient can be used to determine the city.

1NF
Patient Visit (VisitNo, VisitDate,PatNo, PatAge, PatCity, Patzip)
Patient Diagnosis (VisitNo, ProvNo, ProvSpecialty, Diagnosis)

2NF
Patient Visit (VisitNo, VisitDate,PatNo, PatAge, PatCity, Patzip)
Patient Diagnosis (VistitNo, *ProvNo, Diagnosis)
Provider (ProvNo, ProvSpecialty)

3NF
Patient (PatNo, PatAge, *PatZip)
City (PatZip, PatCity)
Patient Visit (VisitNo, VisitDate, *PatNo)
Patient Diagnose (VisitNo, *ProvNo, Diagnosis)
Provider (ProvNo, ProvSpecialty)

7.
//1NF
Corriculum(**CocuCode**, Desc, AdvisorNo, AdvisorName)
Student(**StuID**, **CocuCode*,** StuName, StuContact, day, time)
Curriculum(CoCuCode,Desc,AdvisorNo,AdvisorName)
Registration (*CoCuCode, StuID,StuName,StuContact,Day,Time)

//2NF
Curriculum(CoCuCode,Desc,AdvisorNo,AdvisorName)
Registration (CoCuCode*,StuID*, Day, Time)
Student (StuID,StuName,StuContact)

//3NF
Curriculum(CoCuCode,Desc, AdvisorNo*)
Advisor(AdvisorNo,AdvisorName)
Registration (*CoCuCode, *StuID, Day, Time)
Student (StuID,StuName,StuContact)