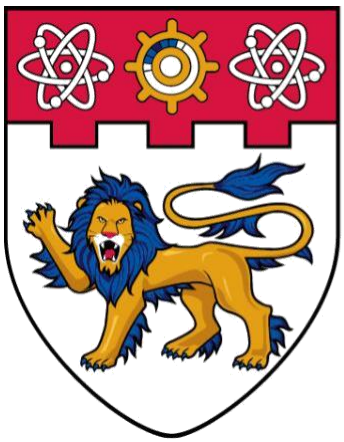

NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

# CZ2007 Introduction to Databases

## Lab 3 Report Submission:
## Generation of Normalized Database Schema


**Name:**

**Aide Iskandar Bin Abdul Rahim (U1922414E)**
**Bryson Teo Yuan Harn (U1920640A)**
**Chien Yong Qiang (U1920129E)**
**Kondreddy Saitejareddy (U1923841F)**
**Pal Aratrika (U1922069F)**
**Ta Quynh Nga (U1920601L)**

**USERS (<u>UID</u>, Name)**
Keys: UID
Primary Key: UID
FDs: UID $\rightarrow$ Name
The relation is in BCNF.

**SHOPS (<u>Sname)</u>**
Keys: Sname
Primary Key: Sname
FDs: None
The relation is in BCNF.

**PRODUCTS (<u>Pname</u>, Maker, Category)**
Keys: Pname
Primary Key: Pname
FDs: Pname $\rightarrow$ Maker, Category
The relation is in BCNF.

Assumption:

- Pname is for privatized products. Privatized products are products with unique name. For example: Macbook Air, Macbook Pro, etc.

**ORDERS (<u>OID</u>, ShippingCost, ShippingAddress, Date-time, UID)**
Keys: OID, {UID, Date-time}
Primary Key: OID
FDs: OID $\rightarrow$ ShippingCost, ShippingAddress, Date-time, UID
{UID, Date-time} $\rightarrow$ OID
{UID, Date-time} $\rightarrow$ ShippingCost, ShippingAddress
The relation is in BCNF.

Assumptions:

- We assume that ShippingCost depends on ShippingAddress, Pname, Sname and other factors so ShippingAddress alone cannot determine ShippingCost.
- We also assume that users may ask for products to be delivered at multiple different addresses so UID cannot determine ShippingAddress.
- We also assume that at one Date-time, a user can click the button and place only one order so UID and Date-time can imply OID.

**COMPLAINTS (<u>CID</u>, Text, Filed-date-time, Status, EID, Handled-date-time, UID)**

Keys: CID, {UID, Filed-date-time}

Primary Key: CID

FDs: CID $\rightarrow$ Text, Filed-date-time, Status, EID, Handled-date-time, UID

{UID, Filed-date-time} $\rightarrow$ CID

{UID, Filed-date-time} $\rightarrow$ Text, Status, EID, Handled-date-time

The relation is in BCNF.

Assumptions:

- We assume EID and Handled-date-time cannot imply the CID because at a specific date and time, a particular employee may handle multiple complaints in a batch if they are of similar nature or are complaining about the same product or shop.
- We assume UID and Filed-date-time can imply the CID because at a specific date and time, a particular user can file only one complaint.

**EMPLOYEES (<u>EID</u>, Ename, Salary)**

Keys: EID

Primary Key: EID

FDs: EID $\rightarrow$ Ename, Salary

The relation is in BCNF.

**PRODUCTS-IN-SHOPS (<u>Pname</u>, <u>Sname</u>, Price, TotalQuantity)**

Keys: {Pname, Sname}

Primary Key: {Pname, Sname}

FDs: {Pname, Sname} $\rightarrow$ Price, TotalQuantity

The relation is in BCNF.

**PRODUCTS-IN-ORDERS (<u>Pname</u>, <u>OID</u>, <u>Sname</u>, Price, Quantity, Status, Delivery-date)**

Keys: {OID, Pname, Sname}

Primary Key: {OID, Pname, Sname}

FDs: {OID, Pname, Sname} $\rightarrow$ Price, Quantity, Status, Delivery-date

The relation is in BCNF.

**COMPLAINTS-ON-SHOPS (<u>CID</u>, Sname)**

Keys: CID

Primary Key: CID

FDs: CID $\rightarrow$ Sname

The relation is in BCNF.

**COMPLAINTS-ON-PRODUCTS (<u>CID</u>, Pname)**
Keys: CID
Primary Key: CID
FDs: CID → Pname
The relation is in BCNF.

**PRICE-HISTORY (<u>Pname</u>, <u>Sname</u>, <u>Start-date</u>, End-date, Price)**
Keys: {Pname, Sname, Start-date}, {Pname, Sname, End-date}
Primary Key: {Pname, Sname, Start-date}
FDs: {Pname,Sname ,Start-date} → Price, End-date
{Pname, Sname, End-date} → Price, Start-date
The relation is in BCNF.

**FEEDBACK (<u>Pname</u>, <u>Sname</u>, <u>OID</u>, UID, Rating, Date-time, Comment)**
Keys: {Pname, Sname, OID}
Primary Key: {Pname, Sname, OID}
FDs: {Pname, Sname, OID} → Rating, Comment, Date-time
OID → UID

The relation violates 3NF because of the OID → UID functional dependency. This functional dependency violates it because OID is not the key/superkey and UID is not a part of a key.

**<u>3NF Decomposition of *FEEDBACK* Relation</u>**
**Step 1: Transform the FDs so that the RHS of every FDs only contains one attribute.**
S= { {Pname, OID, Sname} → Rating,
{Pname, OID, Sname} → Comment,
{Pname, OID, Sname} → Date-time,
OID → UID }

**Step 2: Check for redundant FDs. No redundant FDs.**
S = { {Pname, OID, Sname} → Rating,
{Pname, OID, Sname} → Comment,
{Pname, OID, Sname} → Date-time,
OID → UID }

Note:

- (Sname, Pname, OID $\rightarrow$ Rating) is not redundant as {Sname, Pname, OID}$^+$ = {Sname, Pname, OID, UID, Comment, Date-time} and this isn't true if this FD is removed. This FD cannot be derived from other FDs.
- (Sname, Pname, OID $\rightarrow$ Comment) is not redundant with a similar reasoning as for (Sname, Pname, OID $\rightarrow$ Rating)
- (Sname, Pname, OID $\rightarrow$ Date-time) is not redundant with a similar reasoning as for (Sname, Pname, OID $\rightarrow$ Rating)
- (OID $\rightarrow$ UID) is not redundant as {OID}$^+$ = {OID} if this FD is removed

**Step 3: Check for redundant attributes. No redundant attributes.**

S = { {Pname, OID, Sname}$\rightarrow$ Rating,
{Pname, OID, Sname} $\rightarrow$ Comment,
{Pname, OID, Sname} $\rightarrow$ Date-time,
OID $\rightarrow$ UID }

Note:

- If either of the attributes in the LHS of the FDs is removed, the closure of the remaining attributes would not include the attribute in the RHS. Thus, none of the attributes are redundant.

**Step 4: Merge if the LHS of the FDs have the same attributes.**

S = { {Pname, OID, Sname} $\rightarrow$ Rating, Comment, Date-time,
OID $\rightarrow$ UID }

**Step 5: Create tables for each of the FDs.**

R1(Pname, OID, Sname, Rating, Comment, Date-time)
R2(OID, UID)

**Step 6: Check whether any of the tables contains a key, and remove redundant tables.**

- R1 contains the key {**Pname, Sname, OID}**. Hence, there is no need to create any additional table for the keys.
- There is no redundant table.

**Results:**
**R1(<u>Pname, Sname, OID</u>, Rating, Comment, Date-time)**
**R2(<u>OID</u>, UID)**

Note:

- Table R2(<u>OID</u>, UID) is redundant if we compare it with the "ORDERS" relation.
- Hence we can remove R2 to prevent redundancy since all the data and Functional Dependency from R2 is already available in ORDERS relation.

*FEEDBACK* 3NF Decomposed Relation:
**FEEDBACK(<u>Pname, Sname, OID</u>, Rating, Comment, Date-time)**
Key: {Pname, Sname, OID}
Primary Key: {Pname, Sname, OID}
FDs: {Pname, Sname, OID} $\rightarrow$ Rating, Comment, Date-time
The relation is in both 3NF and BCNF.